

linux内存管理之非连续物理地址分配(vmalloc)Linux认证考试
PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_linux_E5_86_85_E5_AD_c103_644909.htm 前面我们已经分析了linux如何利用伙伴系统,slab分配器分配内存,用这些方法得到的内存在物理地址上都是连续的,然而,有些时候,每次请求内存时,系统都分配物理地址连续的内存块是不合适的,可以利用小块内存“连接”成大块可使用的内存.这在操作系统设计中也被称为“内存拼接”,显然,内存拼接在需要较大内存,而内存访问相比之下不是很频繁的情况下是比较有效的.在linux内核中用来管理内存拼接的接口是vmalloc/vfree.用vmalloc分配得到的内存在线性地址是平滑的,但是物理地址上是非连续的.一:准备知识:Linux用vm_struct结构来表示vmalloc使用的线性地址.vmalloc所使用的线性地址区间为: VMALLOC_START VMALLOC_END.借用It.Understanding.the.Linux.Kernel.3rdgt.中的一副插图,如下示:从上图中我们可以看到每一个vmalloc_area用4KB隔开,这样做是为了很容易就能捕捉到越界访问,因为中间是一个“空洞”.二:相关的数据结构 下面来分析一下vmalloc area的数据结构: struct vm_struct { void *addr. //虚拟地址 unsigned long size. //vm的大小 unsigned long flags. //vm的标志 struct page **pages. //vm所映射的page unsigned int nr_pages. //page个数 unsigned long phys_addr. //对应的起始物理地址 struct vm_struct *next. //下一个vm.用来形成链表 } 全局变量vmlist用来管理vm构成的链表 全局变量vmlist用于访问vmlist所使用的信号量 对于vm_struct有两个常用的操作: get_vm_area/remove_vm_area get_vm_area:用来分配一个合适大小的vm结构,分配成功之后,

将其链入到vmlist中,代码在 mm/vmalloc.c中.如下示: //size为vm的大小 struct vm_struct *get_vm_area(unsigned long size, unsigned long flags) { //在VMALLOC_START与VMALLOC_END找到一段合适的空间 return __get_vm_area(size, flags, VMALLOC_START, VMALLOC_END). } //参数说明: //start:起始地址 end:结束地址 size 空间大小 struct vm_struct * __get_vm_area(unsigned long size, unsigned long flags, unsigned long start, unsigned long end) { struct vm_struct **p, *tmp, *area. unsigned long align = 1. unsigned long addr. //如果指定了VM_IOREMAP.则调整对齐因子 if (flags & VM_IOREMAP_MAX_ORDER) bit = VM_IOREMAP_MAX_ORDER. else if (bit < VM_IOREMAP_MAX_ORDER). //遍历vmlist:找到合适大小的未使用空间 for (p = &tmp; tmp->addr >= start) addr = ALIGN(tmp->addr, align). continue. } //size <= tmp->addr - start goto out. //中间的空隙可以容纳下size大小的vm.说明已经找到了这样的一个vm if (size <= tmp->addr - start) goto found. //调整起始地址为vm的结束地址 addr = ALIGN(tmp->addr, align). //如果超出了范围 if (addr >= start) { *p = tmp; tmp->flags = flags. tmp->size = size. tmp->nr_pages = 0. tmp->vm_list_lock). return tmp. out: 100Test 下载频道开通 , 各类考试题目直接下载。 详细请访问 www.100test.com